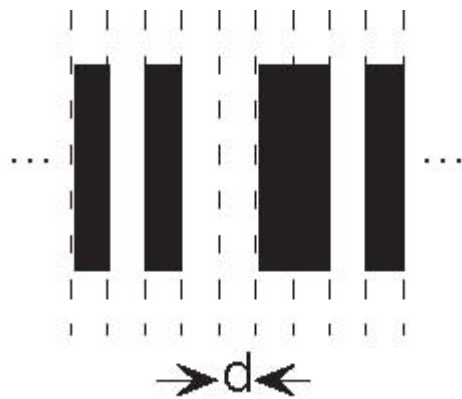


**Table 6.8**

1. What is the average number of bits necessary to represent this alphabet?
2. Using a simple binary code for this alphabet, a two-bit block of data bits naturally emerges. Find an error correcting code for two-bit data blocks that corrects all single-bit errors.
3. How would you modify your code so that the probability of the letter *a* being confused with the letter *d* is minimized? If so, what is your new code; if not, demonstrate that this goal cannot be achieved.

**Problem 6.21: Universal Product Code**

The Universal Product Code (UPC), often known as a bar code, labels virtually every sold good. An example (Figure 6.40) of a portion of the code is shown.

**Figure 6.40**

Here a sequence of black and white bars, each having width  $d$ , presents an 11-digit number (consisting of decimal digits) that uniquely identifies the product. In retail stores, laser scanners read this code, and after accessing a database of prices, enter the price into the cash register.

1. How many bars must be used to represent a single digit?
2. A complication of the laser scanning system is that the bar code must be read either forwards or backwards. Now how many bars are needed to represent each digit?
3. What is the probability that the 11-digit code is read correctly if the probability of reading a single bit incorrectly is  $p_e$ ?
4. How many error correcting bars would need to be present so that any single bar error occurring in the 11-digit code can be corrected?

**Problem 6.22: Error Correcting Codes**

A code maps pairs of information bits into codewords of length 5 as follows.